



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA STROJNÍHO INŽENÝRSTVÍ

FACULTY OF MECHANICAL ENGINEERING

ÚSTAV AUTOMATIZACE A INFORMATIKY

INSTITUTE OF AUTOMATION AND COMPUTER SCIENCE

**NÁVRH A REALIZACE VZOROVÝCH ÚLOH PRO PLC B&R
AUTOMATION**

DESIGN AND IMPLEMENTATION OF TUTORIALS FOR PLC B&R AUTOMATION

BAKALÁŘSKÁ PRÁCE

BACHELOR'S THESIS

AUTOR PRÁCE

AUTHOR

Aleš Vymazal

VEDOUCÍ PRÁCE

SUPERVISOR

Ing. et Ing. Stanislav Lang

BRNO 2017

Zadání bakalářské práce

Ústav: Ústav automatizace a informatiky
Student: **Aleš Vymazal**
Studijní program: Strojírenství
Studijní obor: Základy strojního inženýrství
Vedoucí práce: **Ing. et Ing. Stanislav Lang**
Akademický rok: 2016/17

Ředitel ústavu Vám v souladu se zákonem č.111/1998 o vysokých školách a se Studijním a zkušebním řádem VUT v Brně určuje následující téma bakalářské práce:

Návrh a realizace vzorových úloh pro PLC B&R Automation

Stručná charakteristika problematiky úkolu:

Předmětem bakalářské práce je demonstrace použití automatizačních prostředků společnosti B&R Automation. Student se zaměří na práci s programovatelnými automaty, pro něž vytvoří vzorové řešení vhodně zvoleného automatizačního problému (např. simulace automatické pračky). Student ve své práci rovněž stručně představí dostupné programovací jazyky a nástroje nabízené vývojovým prostředím Automation Studio.

Cíle bakalářské práce:

- Úvod k technologiím B&R Automation a vývojovému prostředí Automation Studio.
- Návrh a realizace jednoduché úlohy (ilustrativního charakteru) v různých programovacích jazycích nabízených prostředím Automation Studio.
- Návrh a realizace úlohy většího rozsahu (např. simulace automatické pračky) v libovolně zvoleném programovacím jazyku.

Seznam doporučené literatury:

ŠVARC, I., ŠEDA, M., VÍTEČKOVÁ, M. Automatické řízení. 1. vyd. Brno, CERM., 2007. 324 s. ISBN 978-80-214-3491-2.

NĚMEC, Z. Prostředky automatického řízení: elektrické. 2. vyd. VUT Brno, 2008.

ZEZULKA, F. Prostředky průmyslové automatizace. VUT Brno, 2004. 161 s.

B&R Automation. [on-line] <http://www.br-automation.com/>

Termín odevzdání bakalářské práce je stanoven časovým plánem akademického roku 2016/17

V Brně, dne

L. S.

doc. Ing. Radomil Matoušek, Ph.D.
ředitel ústavu

doc. Ing. Jaroslav Katolický, Ph.D.
děkan fakulty

ABSTRAKT

Tato bakalářská práce pojednává o firmě B&R Automation a jejích technologiích. Součástí práce je také návod k vývojovému prostředí Automation Studio a stručné srovnání všech dostupných programovacích jazyků tohoto vývojového prostředí. Rozsáhlejší automatizační úloha byla řešena v programovacím jazyce C a testována na B&R PLC X20 CP 1485.

ABSTRACT

The topic of this bachelor's thesis is the company B&R and its technologies. The text contains a tutorial for the Automation Studio development environment and a brief comparison of all programming languages available in Automation Studio. A larger automation task was implemented in the C programming language and tested on B&R PLC X20 CP 1485.

KLÍČOVÁ SLOVA

PLC, B&R, Technologie B&R, Automation Studio, Programovací jazyky, Automatická pračka

KEYWORDS

PLC, B&R, B&R technology Automation Studio, Programming Languages, Automatic Washing machine

BIBLIOGRAFICKÁ CITACE

VYMAZAL, Aleš. *Návrh a realizace vzorových úloh pro PLC B&R Automation*, Brno, 2017. Bakalářská práce. Vysoké učení technické v Brně, Fakulta strojního inženýrství, Ústav automatizace a informatiky.

PODĚKOVÁNÍ

Děkuji svému vedoucímu Ing. et Ing. Stanislavu Langovi za ochotu a pomoc při zpracování bakalářské práce. Dále patří dík mé rodině za podporu a firmě B&R Automation za poskytnuté školení k bakalářské práci.

ČESTNÉ PROHLÁŠENÍ

Prohlašuji, že tato práce je mým původním dílem, zpracoval jsem ji samostatně pod vedením Ing. et Ing. Stanislava Langa a s použitím zdrojů uvedených v seznamu použitých zdrojů.

V Brně dne 26. 5. 2017

.....

Aleš Vymazal

OBSAH

1	ÚVOD.....	15
2	B&R AUTOMATION	17
2.1	Společnost B&R Automation	17
2.2	Technologie firmy B&R.....	17
2.2.1	Řídící technika.....	17
2.2.2	Powerlink.....	18
2.2.3	openSAFETY	18
2.2.4	CNC & Robotika	18
2.2.5	Mapp technologie	19
2.2.6	reACTION technologie	20
2.2.7	Vizualizace	20
3	AUTOMATION STUDIO.....	21
3.1	Úvodní strana a nový projekt.....	21
3.2	Přidání programu a modulů	23
3.3	Konfigurace Ethernetu na cílovém zařízení	24
3.4	Sestavení projektu a vytvoření CompactFlash	25
3.5	Spojení PC a PLC pomocí ethernetu	27
3.6	Testování projektu	28
4	PROGRAMOVACÍ JAZYKY V AUTOMATION STUDIO.....	29
4.1	Textové Jazyky	29
4.1.1	ANSI C a ANSI C++	29
4.1.2	Structured Text (ST).....	30
4.1.3	Automation Basic	30
4.1.4	Instruction List (IL)	31
4.2	Grafické jazyky.....	31
4.2.1	Ladder diagram (LD).....	31
4.2.2	Function Block Diagram (FBD)	32
4.2.3	Continous Function Chart (CFC)	33
4.2.4	Sequential Function Chart (SFC)	34
5	AUTOMATICKÁ PRAČKA	35
5.1	Stručná charakteristika projektu	35
5.2	Popis fází praní	36
5.3	Popis klíčových částí implementace programu.	38
5.3.1	Rotace bubnu	38
5.3.2	Stav Napouštění	39
5.4	Vizualizace projektu	40
6	ZÁVĚR	41
7	SEZNAM POUŽITÝCH ZDROJŮ.....	43
8	SEZNAM PŘÍLOH.....	45

1 ÚVOD

Již několik desítek let pronikají stále více do výrobních procesů automatizované systémy a umělá inteligence. Protože automatizace zvyšuje produktivitu práce, bude pro konkurenceschopnost průmyslu důležitá znalost těchto systémů. V dnešní době se mluví o tzv. 4. průmyslové revoluci, která je založena na principu propojení všech komponent prostřednictvím internetu. Počítačové systémy vybavené umělou inteligencí dokážou řídit celé výrobní procesy, optimalizovat a řešit nepředvídatelné problémy.

Tato bakalářská práce pojednává o vybraných technologiích firmy B&R Automation a vývojovém prostředí Automation Studio. V práci je obsažen postup pro založení projektu v Automation Studio a stručné srovnání všech dostupných programovacích jazyků.

Poslední část práce se zabývá rozsáhlejší automatizační úlohou, konkrétně simulací automatické pračky. V příloze této práce je dodán kompletní projekt a vizualizace zmíněné úlohy.

2 B&R AUTOMATION

2.1 Společnost B&R Automation

B&R Automation je soukromá společnost se sídlem v rakouském městě Eggelsberg specializující se v oblasti průmyslové automatizace. Firma má v současné době po celém světě asi 2500 zaměstnanců působících v 75 zemích a přibližně 200 poboček. Založena byla v roce 1979 Erwinem Berneckerem a Josefem Rainerem a v České Republice působí od roku 1993. Hlavní sídlo firmy v České Republice se nachází v Brně a pobočky jsou také v Praze, Zábřehu a Plzni. B&R uplatňuje své výrobky v mnoha různých oblastech jako je například plastikářský, obalový či textilní průmysl, energetika, těžba ropy a plynu či automobilový průmysl. Mezi nejvýznamnější produkty B&R patří průmyslové PC a vizualizační prostředky, které poskytují přehled o strojích a systémových stavech, jednotný software pro automatizační projekty (Automation Studio) nebo také kontrolní systémy a mnoho dalších.[1]



Obr.1 Sídlo firmy v Eggelsburgu [2]

2.2 Technologie firmy B&R

2.2.1 Řídící technika

Řídící systémy od B&R jsou využívány pro zvýšení produktivity strojů a systémů. Odolné strategie řízení zajišťují vysokou přesnost strojů a konzistentní kvalitu vyráběného zboží. Všechny řídicí prvky jsou k dispozici jako knihovny pro software Automation Studio a jejich použití je velmi jednoduché. Při implementaci do stroje nebo systému se tedy výrazně zkracuje čas při vypracování projektu. B&R implementovalo do Automation Studio řídicí prvky, které se využívají například ve větrných turbínách, hydraulice, měření teploty, pro pohonné systémy apod. [1]

2.2.2 Powerlink

Ethernet Powerlink je průmyslový komunikační protokol, který byl uveden na trh v roce 2001 firmou B&R. Ethernet Powerlink vznikl za účelem vybudovat rychlou sběrnici s deterministickými odezvami a s minimálním rozptylem časování telegramů. Systém Ethernet Powerlink vychází ze standardu Ethernet, nevyžaduje žádný speciální hardware a využívá všechny standardy Ethernetu, a proto lze pro něj využít všechny systémy určené pro Ethernet. Přenosová rychlost Ethernet Powerlink je až 100 Mb/s a přesnost synchronizace dosahuje zhruba 100 ns, což umožňuje zkombinovat do jedné sítě nejnáročnější úkoly robotiky, řízení nebo CNC. Standard Ethernet Powerlink používá komunikační model producent/konzument, což vede k zvětšení výkonu a propustnosti sítě. [1][3][4]

2.2.3 openSAFETY

Bezpečnostní prvky jako například kabeláž či bezpečnostní relé se dnes postupně nahrazují programovatelnými systémy, které jsou v případě potřeby snadno modifikovatelné. Použití technologie openSAFETY má spoustu výhod. Kombinace rychlejších časů odezvy s menší bezpečnostní vzdáleností zvyšuje produktivitu, což je významné zejména pro výrobce strojů. OpenSAFETY eliminuje chyby a závady, velmi dobře sdílí informace mezi řídicím a bezpečnostním systémem, zjednodušuje bezpečnostní nastavení a chytře reaguje na nebezpečné stavy. Protokol OpenSAFETY je nezávislý na typu sběrnice, lze ho použít s různými sběrnici včetně průmyslového Ethernetu.

B&R nabízí s využitím openSAFETY úplný přehled nad všemi vstupními/výstupními kanály v jednom systému. To umožňuje bezpečné uvedení stroje do provozu pouhou konfigurací v počítači. [1][3]

2.2.4 CNC & Robotika

Průmyslové roboty najdeme v mnoha odvětvích průmyslu. Efektivita výroby díky nim stoupá kvůli snížení materiálových ztrát na minimum. Z ekonomického hlediska jsou roboty obzvláště výhodné pro výrobu velkých sérií, protože se omezují vedlejší neproduktivní časy a zvyšuje se výrobní kapacita.

B&R vylepšují účinnost systémů a strojů kombinací řízení CNC a robotiky v jedné softwarové platformě GMC (Generic Motion Control). GMC lze použít pro všechny typy pohonů a je snadno integrováno do aplikací stroje. [1][5]



Obr. 2 CNC & Robotika [1]

2.2.5 Mapp technologie

Mapp je aplikace, se kterou je možno vytvářet software pro jakýkoliv typ systému či stroje. Mapp obsahuje několik komponent, přičemž každá poskytuje určitou funkci. Hlavní výhodou mapp je, že šetří programátorům čas a náklady díky svým funkčním blokům, které nahrazují základní funkce. Mapp komponenty jsou včleněny do vývojového prostředí Automation Studio od B&R. Tyto komponenty umožňují například uvést do provozu víceosý systém s vačkovými profily, elektronickými převody nebo různými robotickými kinematikami, o to v řádu několika hodin. [1][6]

Vybrané funkce mapp jsou například:

- mapp Robox

Kontroluje jakoukoliv kinematiku stroje. Vývojáři mají možnost ovládat až patnáct os najednou. Robot je snadno nastavitelný díky vizualizaci na panelu. [6]

- mapp AlarmX

Tato funkce zvládá spouštět varovná hlášení centrálně, proto není nutné programovat všechna chybová hlášení jednotlivě. Mapp AlarmX je schopno spustit PDF soubor, přehrát instruktážní video nebo zobrazit model stroje, zdůrazňující jeho chybu. [6]

- mapp OEE

mapp OEE počítá celkovou účinnost stroje a zobrazuje ji na uživatelském rozhraní. Vykresluje grafy ukazující efektivitu, což usnadňuje zvýšení produktivity. [6]



Obr. 3 Ukázka rozhraní mapp technologie [3]

2.2.6 reACTION technologie

Díky technologii reACTION dokáže firma B&R snížit reakční dobu v aplikacích průmyslové automatizace až na 1 μ s. Tento nový přístup dovoluje řídit extrémně časově kritické procesy s použitím standardního hardware a snížit tak náklady snížením vytížení procesoru a optimalizací výkonu ke splnění požadavků nejnáročnějších strojů např. v textilním, obalovém či plastikářském průmyslu. Jeden z praktických příkladů může být dávkování lepidla v obalovém průmyslu. [7]

Aby mohly balicí stroje vyrábět vysoce kvalitní produkty, musejí při práci s lepenkou nebo jinými tuhými obalovými materiály aplikovat tečky lepidla dle extrémně přesných vzorů. Stroj, který produkuje 600 blistrů za minutu má dobu synchronizace nižší než 1 milisekundu. S technologií reACTION lze tečkový vzor synchronizovat do polohy snímače za méně než 200 nanosekund. [1]

2.2.7 Vizualizace

Vizualizace využívá grafické zobrazení procesu a snaží se ovládání centralizovat na jedno místo. Od desek s tlačítky a signalizačními žárovkami se vizualizace vyvinula až do stavu, kdy máme možnost řídit proces centrálně jedním průmyslovým počítačem nebo PLC a vizualizaci můžeme zobrazovat na plochých obrazovkách s dotykovou vrstvou, případně různými tlačítky. [8]

B&R nabízí tato zařízení, která disponují odolností a životností. Ovládání je zajištěno buď přes displej, funkčními klávesy nebo kombinací obou. Uživatel má přehled o systémech a strojích na širokoúhlé obrazovce, mobilním panelu nebo terminálu. [1]



Obr. 4 Panel vizualizace [1]

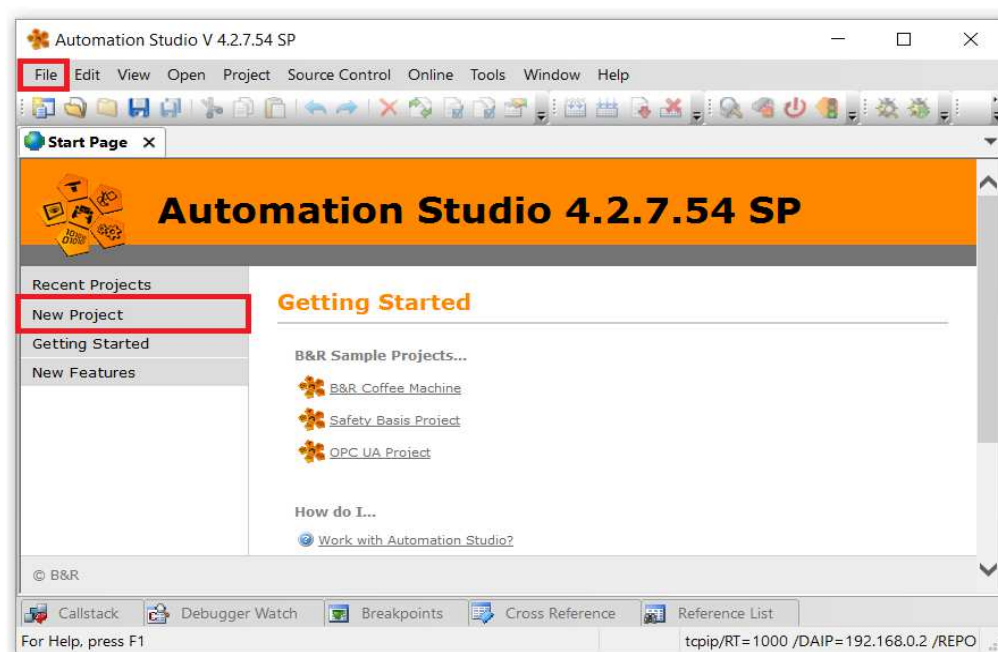
3 AUTOMATION STUDIO

Automation Studio je integrované vývojové prostředí od B&R pro jakékoli automatizační řešení – od řídicí a pohonné techniky až po HMI, vizualizaci, provoz a integrovanou bezpečnostní techniku.

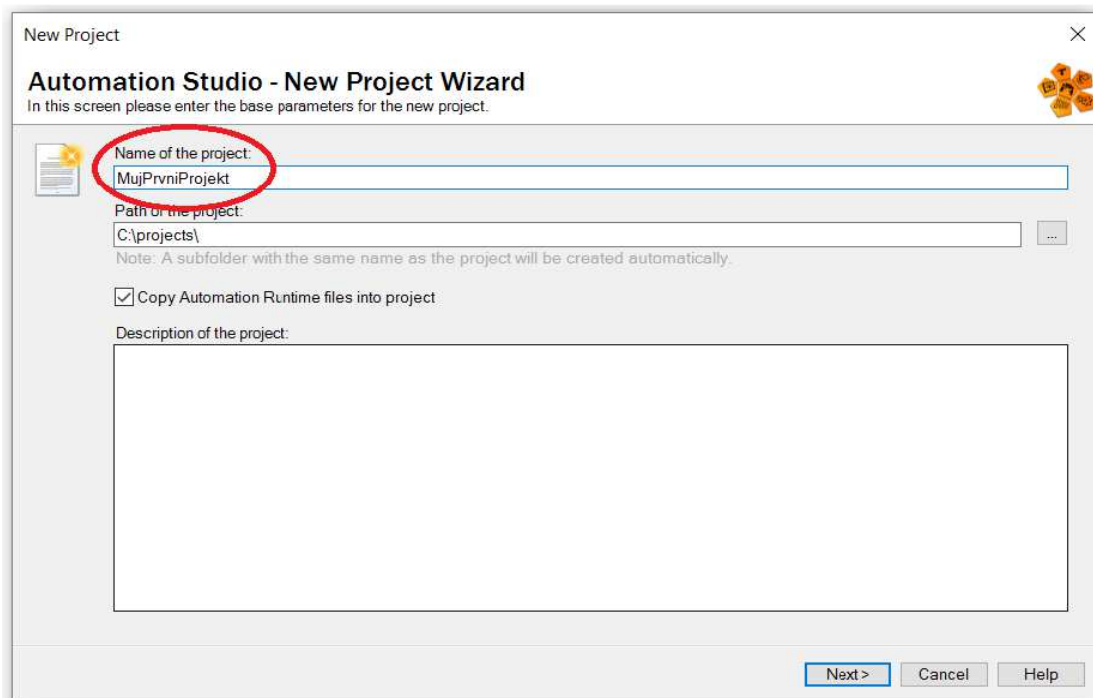
3.1 Úvodní strana a nový projekt

Po nainstalování Automation Studia a prvním spuštění se nám zobrazí úvodní strana, kde najdeme ukázkové projekty od B&R nebo užitečné odkazy na nápovědu.

Nový projekt vytvoříme přes odkaz **New Project** a poté klikneme na **An empty Project** nebo na horní liště pod záložkou **File** se nám v nabídce objeví jako první **New Project**.

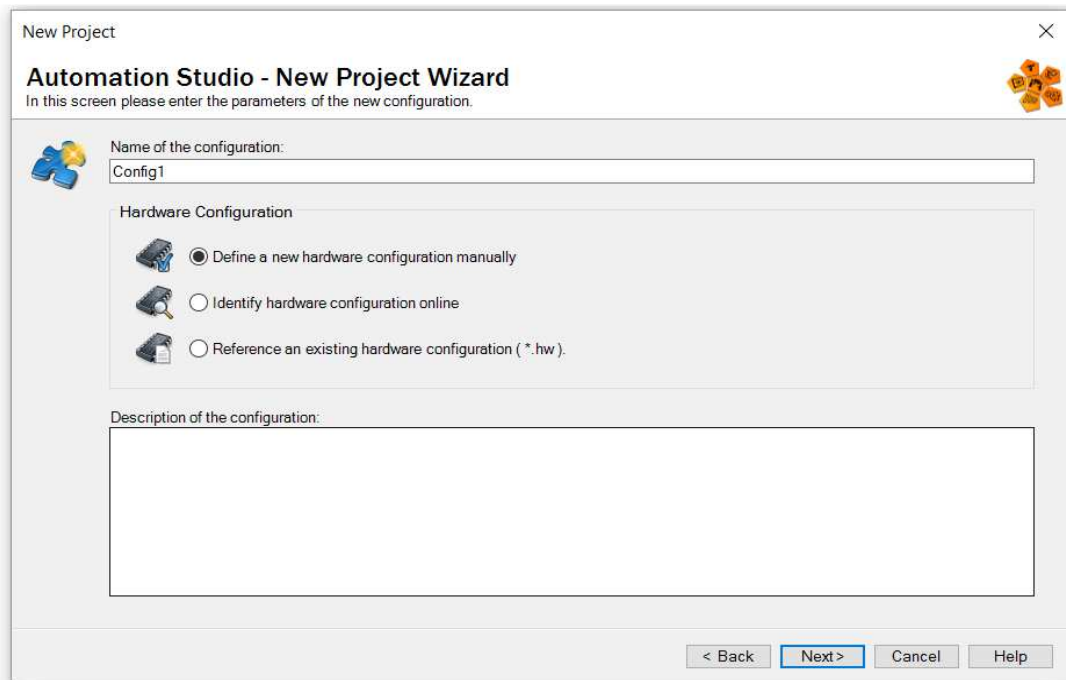


Pojmenujeme projekt, zvolíme, kde má být uložen (automaticky je přednastaveno C:\projects\)) a pokud budeme chtít, můžeme dát projektu krátký popis.



The image shows the 'Automation Studio - New Project Wizard' dialog box. The title bar says 'New Project'. The main title is 'Automation Studio - New Project Wizard'. Below the title, it says 'In this screen please enter the base parameters for the new project.' There are three input fields: 'Name of the project:' with the text 'MujPrvniProjekt', 'Path of the project:' with the text 'C:\projects\', and a checkbox labeled 'Copy Automation Runtime files into project' which is checked. Below these is a large text area labeled 'Description of the project:'. At the bottom right, there are three buttons: 'Next >', 'Cancel', and 'Help'.

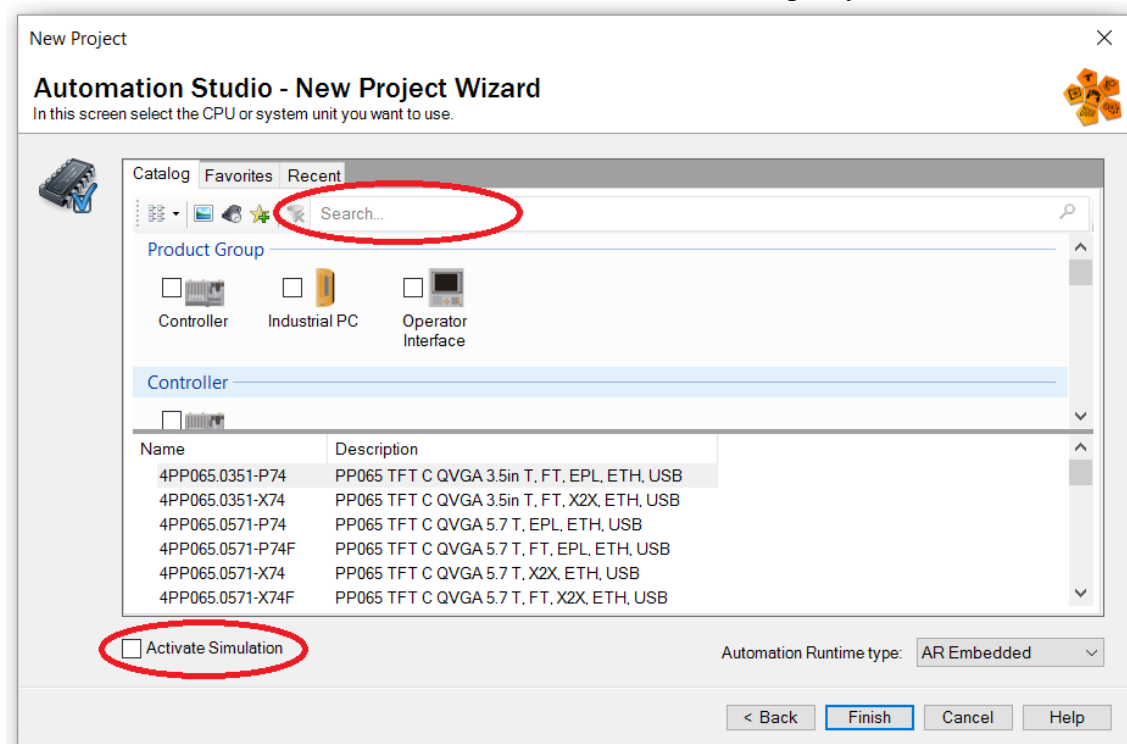
V dalším kroku můžeme změnit jméno naší konfigurace a způsob sestavení hardwaru. V našem případě ponecháme **Define a new hardware configuration manually** a pokračujeme kliknutím na ikonu **Next**.



The image shows the 'Automation Studio - New Project Wizard' dialog box, step 2. The title bar says 'New Project'. The main title is 'Automation Studio - New Project Wizard'. Below the title, it says 'In this screen please enter the parameters of the new configuration.' There are three input fields: 'Name of the configuration:' with the text 'Config1', 'Hardware Configuration' section with three radio buttons: 'Define a new hardware configuration manually' (selected), 'Identify hardware configuration online', and 'Reference an existing hardware configuration (*.hw)', and a large text area labeled 'Description of the configuration:'. At the bottom right, there are four buttons: '< Back', 'Next >', 'Cancel', and 'Help'.

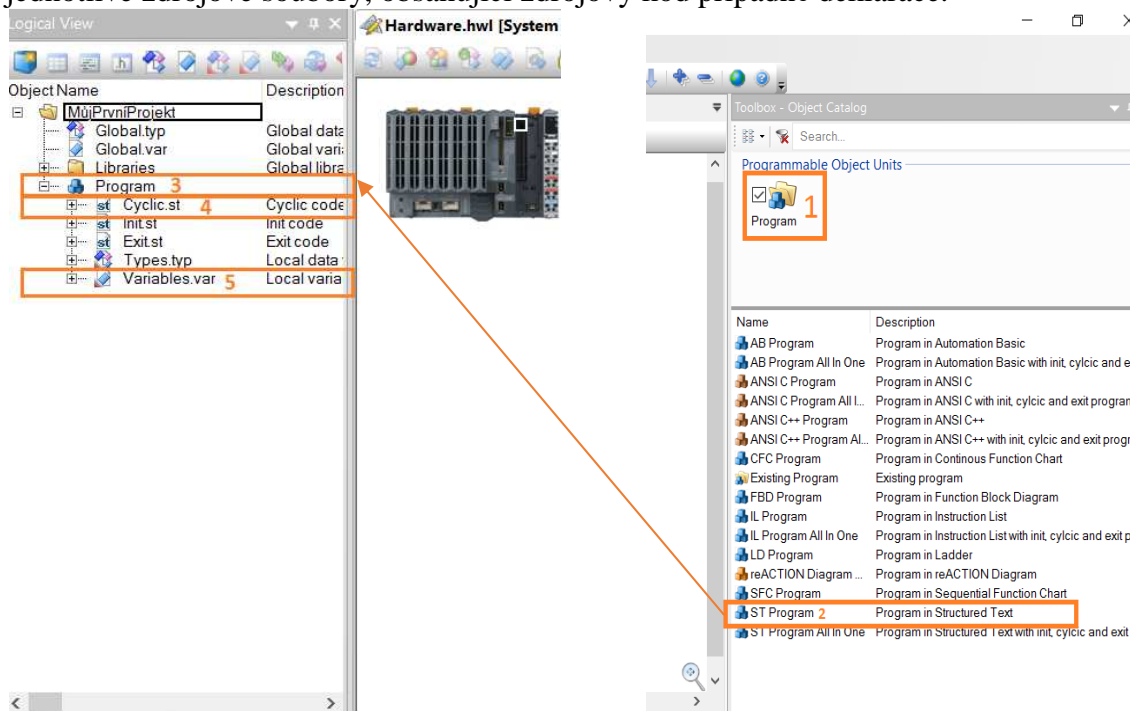
Vybereme náš hardware, pokud známe jeho přesný název, napíšeme jej do kolonky **Search** a dokončíme ikonou **Finish**. Jestliže neznáme název, lze jej dohledávat v seznamu. Při označení některé z ikon v Product Group se nám zúží výběr.

Při označení **Activate Simulation** bude simulátor zapnutý hned od začátku.

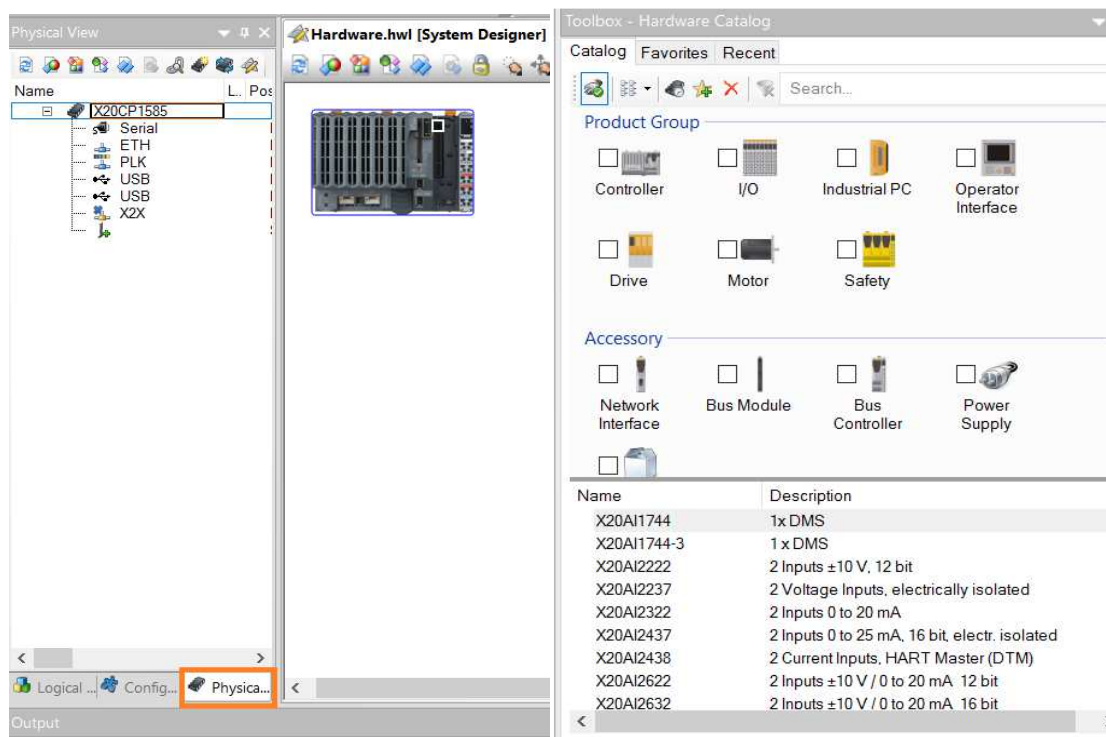


3.2 Přidání programu a modulů

Program přidáme označením ikony **Program**, vybereme programovací jazyk, ve kterém chceme programovat a dvojklikem na něj se nám objeví v **Logical View**. Po otevření našeho Programu v Logical View se objeví jeho části. Vložený objekt zahrnuje jednotlivé zdrojové soubory, obsahující zdrojový kód případně deklarace.

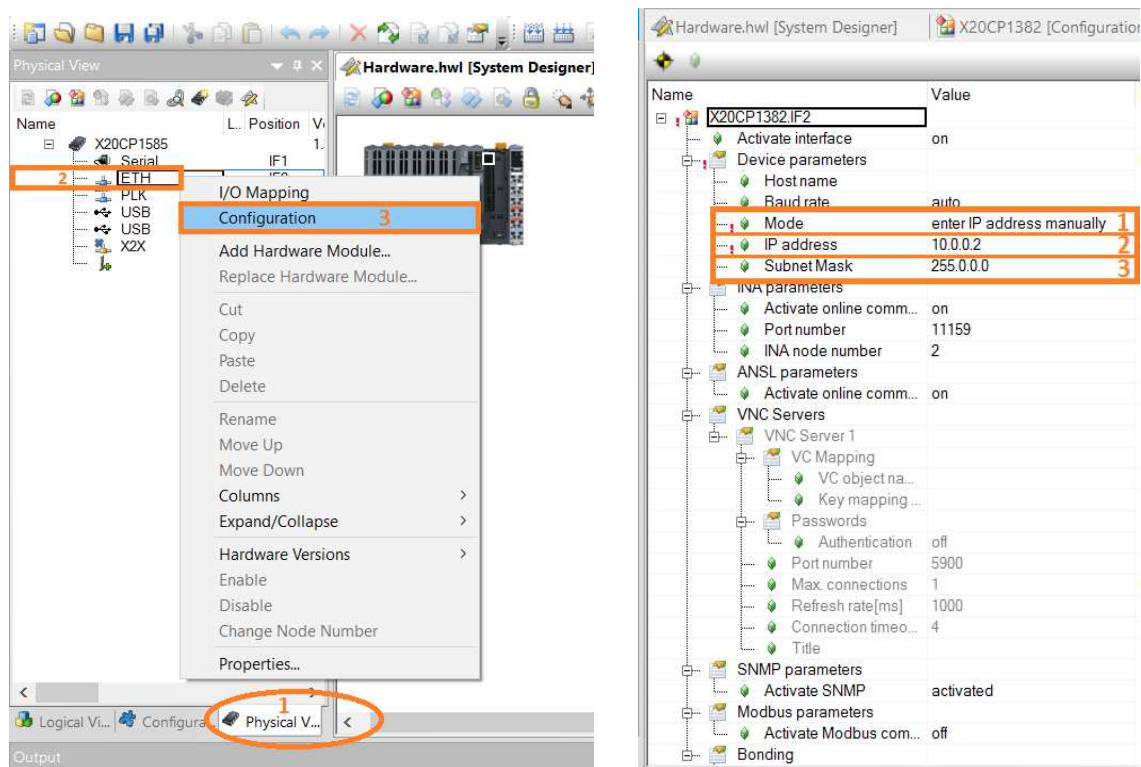


Po kliknutí na **Physical View** se nám objeví **Hardware Catalog**. Vybereme moduly a sestavíme je tak, jak jsou u našeho PLC zapojeny.



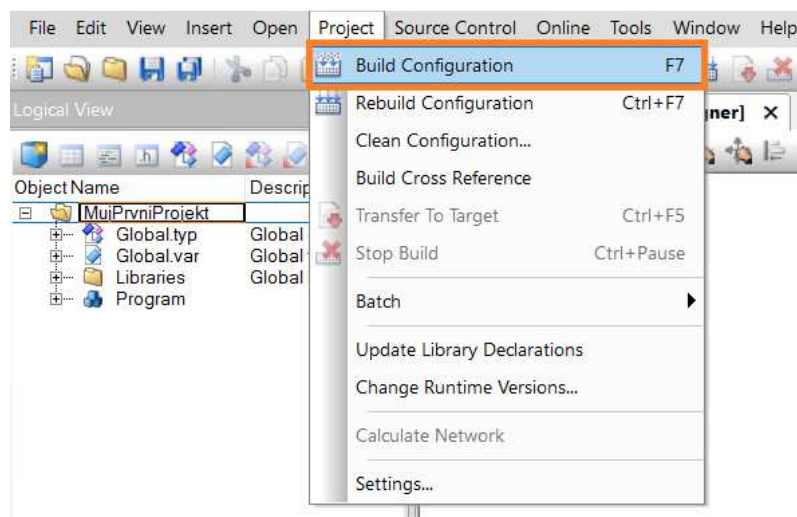
3.3 Konfigurace Ethernetu na cílovém zařízení

Ve sloupci **Physical View** pravým tlačítkem klikneme na **ETH** a vybereme **Configuration**. Objeví se okno pro konfiguraci našeho hardware. Dvakrát klikneme na **Mode** a zvolíme **enter IP address manually**, napíšeme IP adresu našeho PLC a upravíme masku podsítě, podle naší potřeby (nebo necháme nastavení dle DHCP)

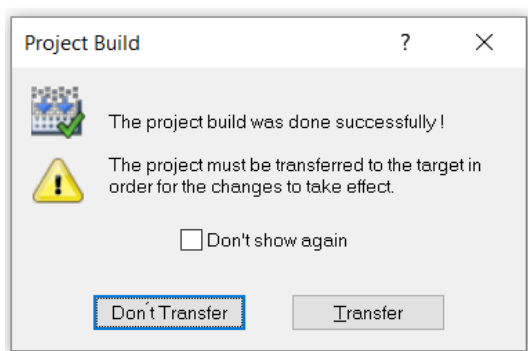


3.4 Sestavení projektu a vytvoření CompactFlash

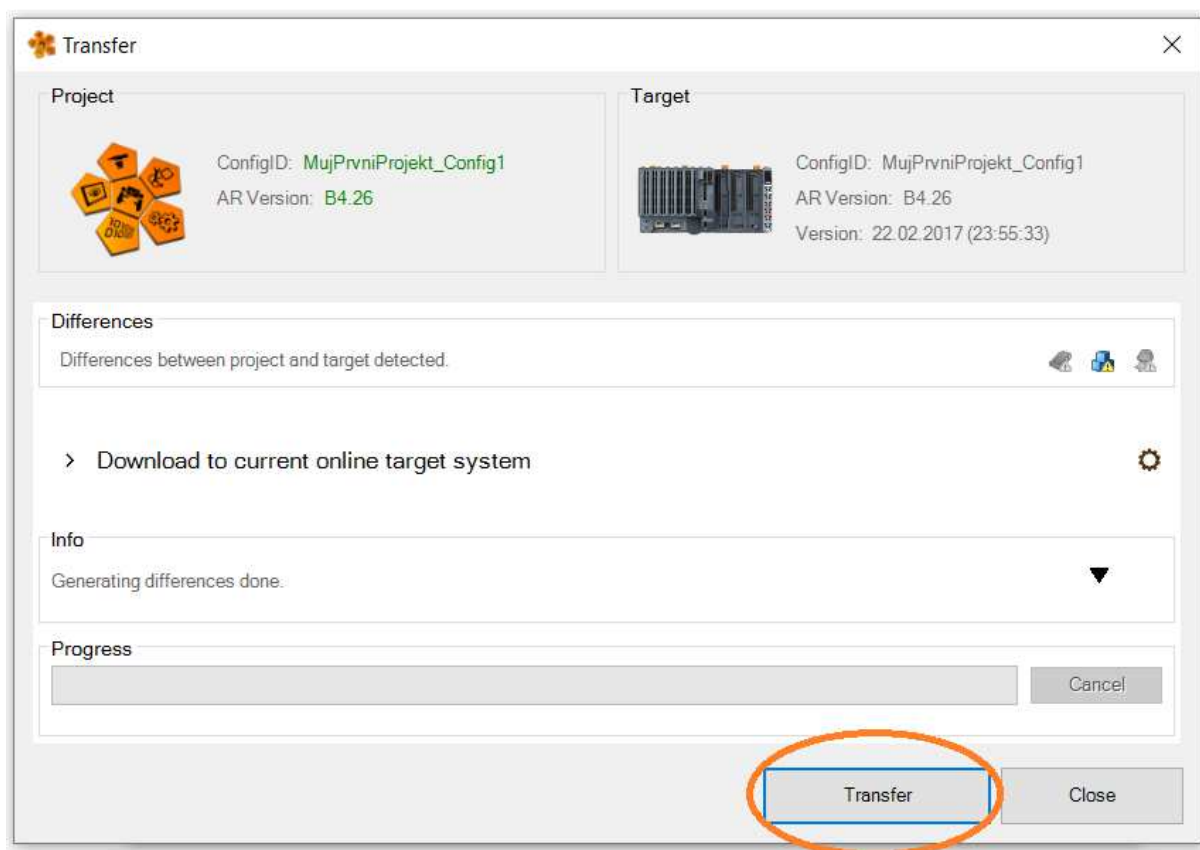
Na horní liště otevřeme záložku Project, kliknutím na Build Project se nám projekt zkompiluje.



Pro vytvoření **CompactFlash** pokračujeme kliknutím na **Transfer**.



Kliknutím na **Transfer** se vytvoří **CompactFlash** se všemi potřebnými komponenty a programy.

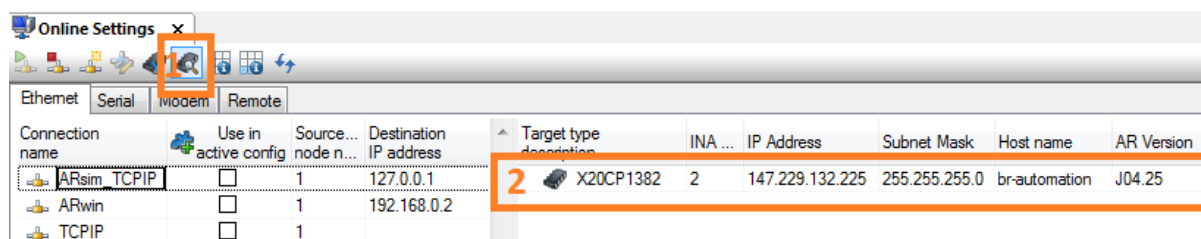


3.5 Spojení PC a PLC pomocí ethernetu

a) Připojení PLC prostřednictvím sítě

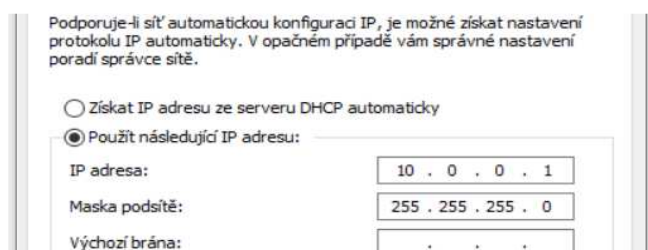
V případě, že jsme v dosahu sítě, je možné on – line spojení. Podmínkou pro on – line spojení je potřeba, aby naše síť podporovala službu SNMP (protokol pro monitorování a správu sítě a služeb).

Na horní liště klikneme na **Online** a vybereme **Settings**, dále pokračujeme ikonou **Browse** a dvojklikem se připojíme k námi zvolenému PLC.

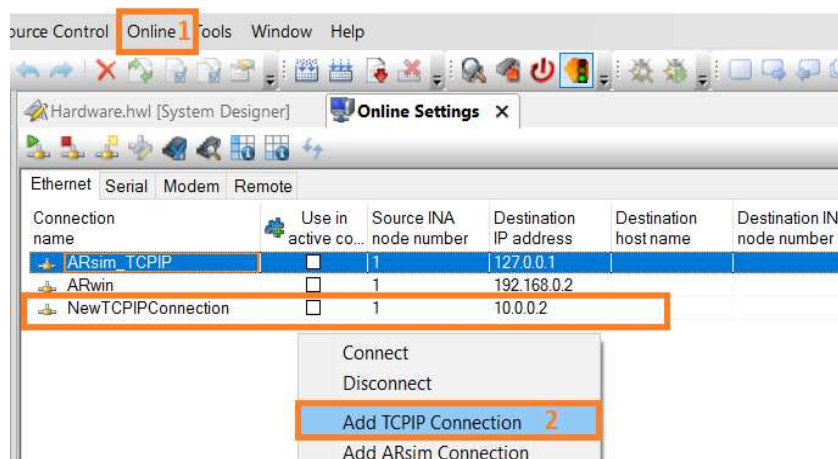


b) Přímé propojení PLC s PC

Abychom spojili PC a PLC pomocí Ethernetu, musíme nastavit **IP adresu** a **masku podsítě** našeho počítače. K požadovanému nastavení našeho počítače se dostaneme ve Windows touto cestou: **Ovládací panely** → **Sít' a internet** → **Centrum síťových připojení a sdílení** → v levém sloupci **Změnit nastavení adaptéru** → **Sít' a Ethernet** → **Protokol IP verze 4 (TCP/IPv4)**.

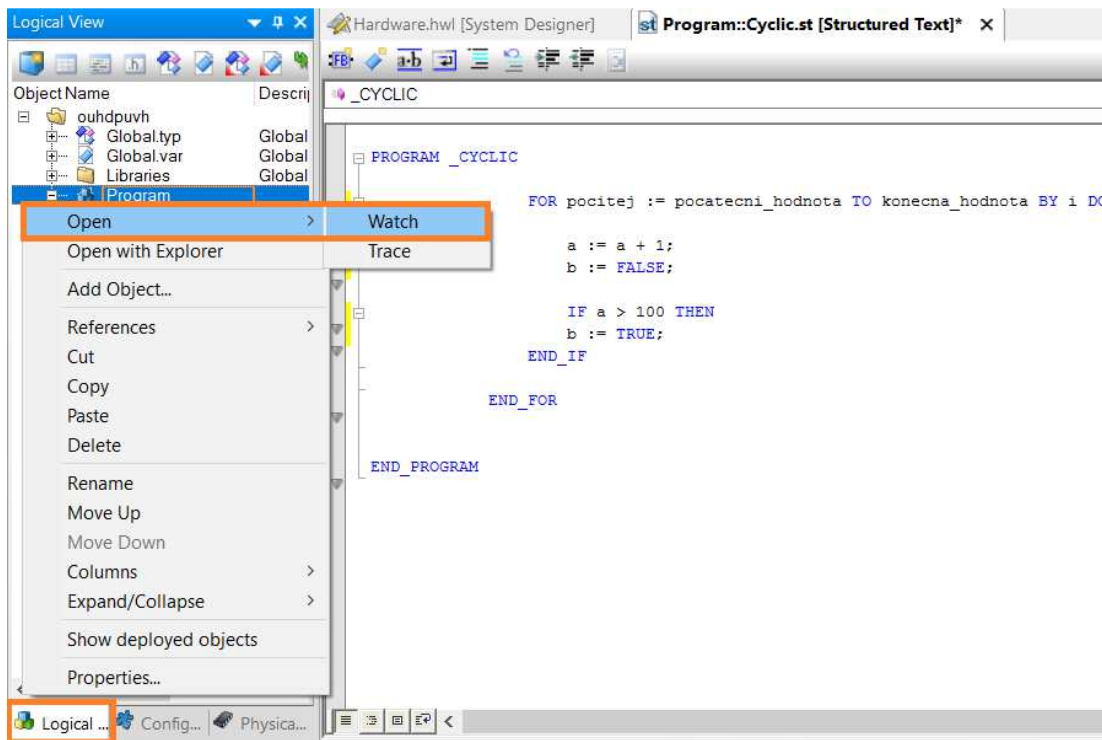


Pro online spojení s našim PLC se vrátíme do Automation Studia, na horní liště vybereme záložku **Online** a otevřeme **Settings** a nastavíme IP adresu našeho PLC ve sloupci **Destination IP address**. Tato IP adresa se musí shodovat s IP adresou, kterou jsme nastavovali při konfiguraci ethernetu.

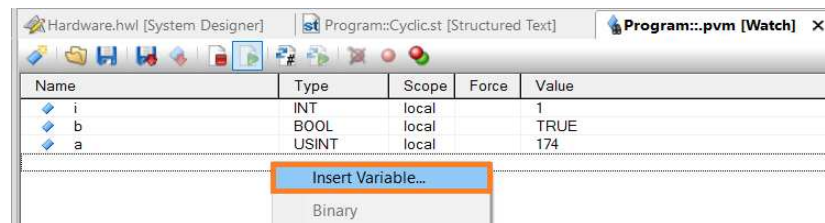


3.6 Testování projektu

Pro sledování hodnot v reálném čase slouží diagnostický nástroj **Watch**. Přepneme se do **Logical View** a pravým tlačítkem klikneme na **Program** → **Open** → **Watch**.



Otevře se nám prázdné okno, klikneme kdekoli pravým tlačítkem a zvolíme **Insert Variable**. Označíme proměnné, které chceme sledovat a přidáme je pomocí **Add**.



4 PROGRAMOVACÍ JAZYKY V AUTOMATION STUDIO

K demonstraci programovacích jazyků byla vytvořena jednoduchá vzorová úloha *Osvětlení poschodí* ve všech dostupných programovacích jazycích Automation Studia.

Při průchodu poschodím se automaticky zapne osvětlení. Když čidlo zaznamená pohyb, osvětlení se rozsvítí na 10 sekund, pokud trvá průchod déle, tak se osvětlení zapne na další časový interval 10 sekund, dokud čidlo nepřestane zaznamenávat pohyb.

4.1 Textové Jazyky

Umožňují práci s daty, textovými řetězci, databázemi a jsou vhodné pro složitější algoritmy. Pro obsluhu a začátečníky jsou však méně čitelné, je nutno znát jejich syntax. K standardizovaným jazykům podporovaným normou IEC 61131-3 patří jazyky Structured Text a Instruction List.

4.1.1 ANSI C a ANSI C++

V Automation Studiu je možno využít všechny dostupné funkce programovacího jazyka C a dokonce i objektově orientovaného programování v jazyce C++.

Největší výhodou těchto jazyků je jejich rozšířenost mezi klasickými programátory, zároveň se hodí pro tvorbu rozsáhlých aplikací a knihoven. Na druhou stranu pro začátečníky jsou tyto jazyky ze všech dostupných ty nejtěžší. [10]

```
void _CYCLIC ProgramCyclic(void)
{
    if(Cidlo == 1)          //V případě že čidlo zaznamená pohyb, zapne se osvětlení
    {
        Osvetleni = 1;
    }
    if(Osvetleni == 1)      //Na proměnou Osvetleni je navázán výstup v automatu
    {
        Casovac.IN = 1;          //Spuštění časovače
        Casovac.PT = 10000;      //Nastavení časovače na 10s
        KonecIntervalu = Casovac.Q;
        if(KonecIntervalu == 1)  //Po 10s se časovač vypne a hodnoty
                                //Osvetleni a KonecIntervalu se vynulují
                                //pro další průchod
        {
            Osvetleni = 0;
            KonecIntervalu = 0;
            Casovac.IN = 0;
        }
    }
    TON(&Casovac);
}
```

Obr.1 ANSI C – Průchod poschodím

4.1.2 Structured Text (ST)

Je výkonný vyšší programovací jazyk, který má kořeny v jazycích Pascal a C. Syntax a obzvlášť struktura programu jsou velmi podobné jako právě v Pascal nebo C. Dodnes je to nejrozšířenější textový programovací jazyk pro PLC. [10]

```
PROGRAM _CYCLIC

    IF Cidlo THEN                //V případě, že čidlo zaznamená pohyb, zapne se osvětlení
        Osvetleni := 1;
    END_IF

    IF Osvetleni THEN            //Na proměnou Osvetleni je navázán výstup v automatu
        Casovac.IN := 1;        //Spuštění časovače
        Casovac.PT := T#10s;    //Nastavení časovače na 10s

        KonecIntervalu := Casovac.Q;

        IF KonecIntervalu THEN    //Po 10s se časovač vypne a hodnoty
                                   //Osvetleni a KonecIntervalu se vynulují pro další průchod
            Osvetleni := 0;
            KonecIntervalu := 0;
            Casovac.IN := 0;
        END_IF
    END_IF
    Casovac();

END_PROGRAM
```

Obr.2 Structured Text – Průchod poschodím

4.1.3 Automation Basic

Programovací jazyk od B&R Automation. Ve srovnání s předcházejícími dvěma jazyky má nejjednodušší syntax, ale jeho použití a výhody jsou prakticky totožné.[1]

```
PROGRAM _CYCLIC

    IF Cidlo THEN                //V případě že čidlo zaznamená pohyb, zapne se osvětlení
        Osvetleni = 1
    ENDIF

    IF Osvetleni THEN            //Na proměnou Osvetleni je navázán výstup v automatu
        Casovac.IN = 1          //Spuštění časovače
        Casovac.PT = T#10s      //Nastavení časovače na 10s
        KonecIntervalu = Casovac.Q

        IF KonecIntervalu THEN    //Po 10s se časovač vypne a hodnoty
                                   //Osvetleni a KonecIntervalu se vynulují pro další průchod
            Osvetleni = 0
            KonecIntervalu = 0
            Casovac.IN = 0
        ENDIF
    ENDIF
    Casovac FUB TON()

END_PROGRAM
```

Obr.3 Automation Basic – Průchod poschodím

4.1.4 Instruction List (IL)

Patří do skupiny textových jazyků, avšak od třech předcházejících se liší. Program je složený z posloupnosti instrukcí (např. sečtení dvou hodnot, přiřazení apod.). Každá instrukce začíná na novém řádku a obsahuje operátor, v případě potřeby modifikátor a jeden nebo více operandů.

Tento programovací jazyk umožňuje rychlé vykonávání programu a je nenáročný na velikost. Zápis se stává při programování složitějších algoritmů nepřehledný a je nutno znát spoustu příkazů, proto se nehodí pro složitější aplikace.[10][11][12]

```
PROGRAM _CYCLIC

LD Cidlo                //V případě že čidlo zaznamená pohyb, zapne se osvětlení
S Osvetleni
LDN Osvetleni
JMPC Label_1
LD 1
ST Casovac.IN           //Spuštění časovače
LD T#10s                //Nastavení časovače na 10s
ST Casovac.PT
LD Casovac.Q
ST KonecIntervalu      //Po 10s se časovač vypne a hodnoty
R Osvetleni             //Osvětlení a KonecIntervalu se vynulují pro další průchod*/
R KonecIntervalu
R Casovac.IN
Label_1:
CAL Casovac()

END_PROGRAM
```

Obr.4 Instruction List – Průchod poschodím

4.2 Grafické jazyky

Hodí se zejména pro kratší programy. Poskytují přehled a snadnou orientaci. Nejsou vhodné pro programování složitých a rozsáhlejších aplikací. Práce s ASCII řetězci je komplikovaná. K standardizovaným jazykům podporovaným normou IEC 61131-3 patří jazyky Ladder Diagram a Function Block Diagram.

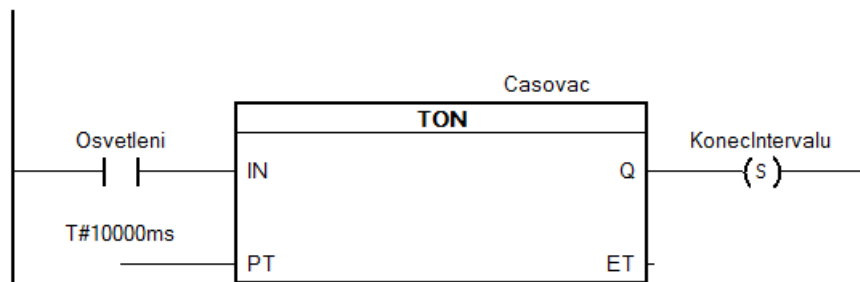
4.2.1 Ladder diagram (LD)

Je založen na grafické reprezentaci reléové logiky. Síť v jazyku LD je zleva i zprava ohraničena svislými čarami, které se nazývají levá a pravá napájecí sběrnice. Mezi těmito sběrnicemi se mohou nacházet kontakty, které představují bitové operace (např. OR, NOT, XOR) a složitější příkazy typu aritmetická operace, časovač nebo přesun dat, které jsou zobrazeny pomocí obdélníků s vývody. Ladder je výhodný při zpracování velkého množství vstupů a výstupů. [10][12]

PROGRAM_CYCLIC

0001

Nastavení časovače na 10s



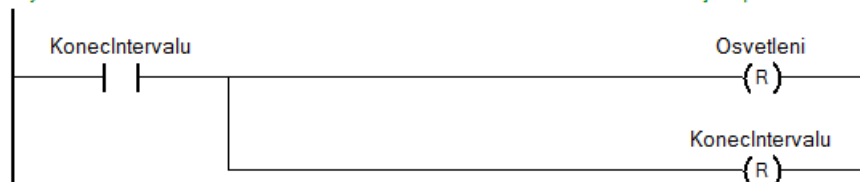
0002

V případě, že čidlo zaznamená pohyb, zapne se osvětlení



0003

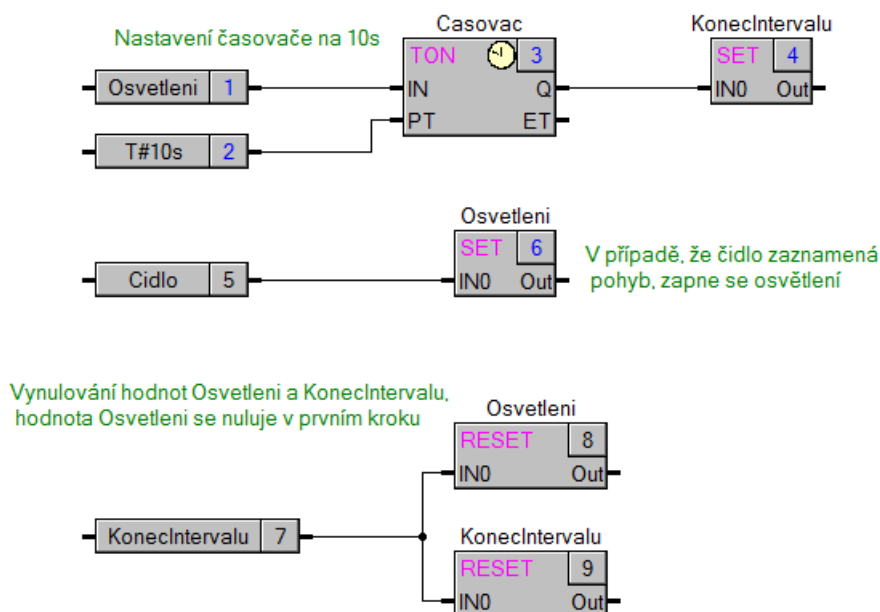
Vynulování hodnot Osvetleni a KonecIntervalu, hodnota Osvetleni se nuluje v prvním kroku



Obr.5 Ladder Diagram – Průchod poschodím

4.2.2 Function Block Diagram (FBD)

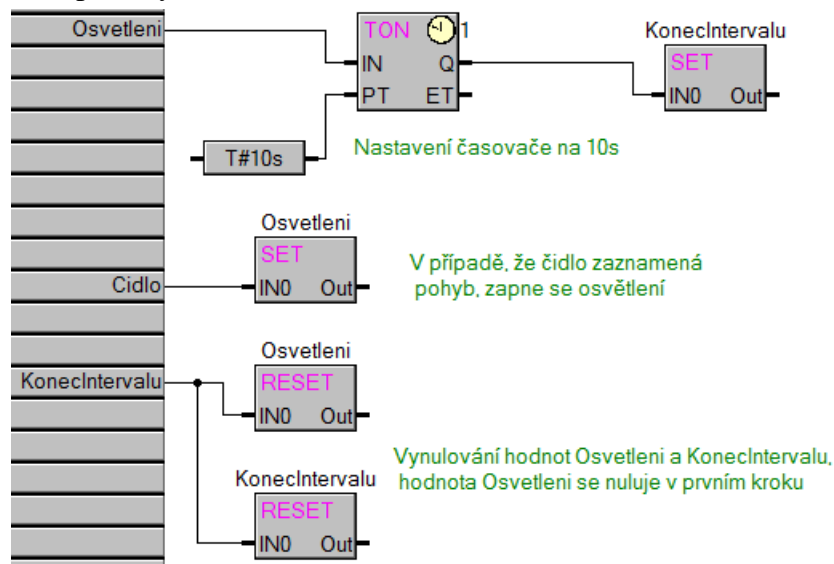
Pracuje s funkčními bloky, přičemž každý blok realizuje určitou funkci (např.logickou či aritmetickou operaci, čítač, časovač apod.). Funkční bloky jsou za sebou liniově propojeny. FBD je podobné LD s tím rozdílem, že se zde nepoužívá reléový zápis. Rovněž jako Ladder je FBD výhodný při zpracování velkého množství vstupů a výstupů. [9]



Obr.6 Function Block Diagram – Průchod poschodím

4.2.3 Continous Function Chart (CFC)

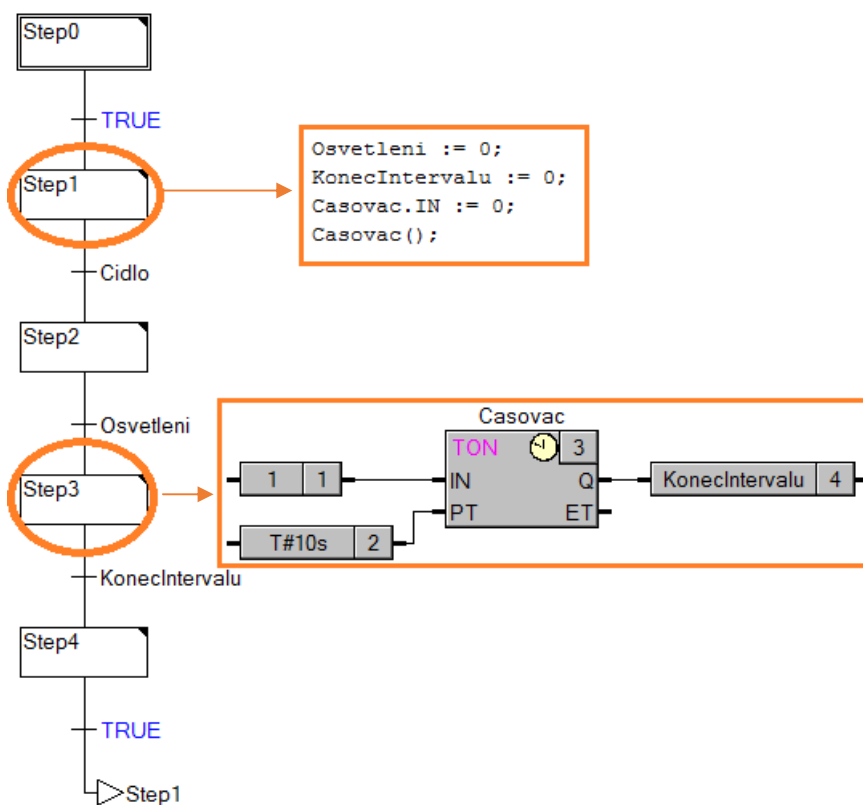
Funkční bloky zde reprezentují jednotlivé funkce, stejně jako u FBD. Bloky jsou pak vzájemně propojeny čarou přes vývody a je zde možnost vytvářet složené funkční bloky. CFC se liší od FBD a LD tím, že se zde pracuje na volné ploše a bloky můžeme uspořádat podle naší potřeby. [10]



Obr.7 Continous Function Chart – Průchod poschodím

4.2.4 Sequential Function Chart (SFC)

Funguje na principu vývojového diagramu. Jednotlivé funkční bloky se dají otevřít a naprogramovat ve kterémkoliv programovacím jazyce nabízeným Automation Studio s výjimkou C a C++. [10]



Obr.8 Sequential Function Chart – Průchod poschodím

5 AUTOMATICKÁ PRAČKA

5.1 Stručná charakteristika projektu

Cílem projektu automatická pračka bylo napsat program do automatické pračky v libovolném programovacím jazyce nabízeným Automation Studio. Pro inspiraci byl použit skutečný prací cyklus moderní automatické pračky LG model WM3250HWA. Prací pračka má několik fází praní, přičemž každá fáze má své stavy. Projekt byl řešen v programovacím jazyce C a pro přehlednost byly využívány strukturované a výčtové datové typy.

Projekt má dva programy – *Pracka* a *Simulator*, v programu *Simulator* jsou simulovány stavy napouštění, vypouštění a sypání pracího prášku. Program *Pracka* obsahuje celý prací cyklus.

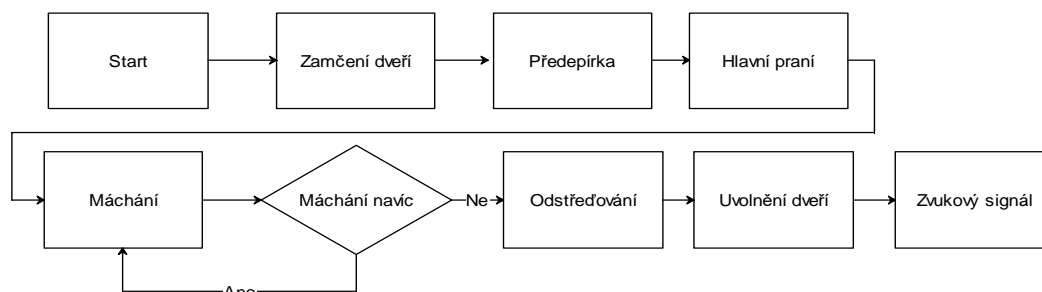
Pro sledování jednotlivých cyklů a fází byl použit diagnostický nástroj *Watch*, který zobrazuje vnitřní stavy modelu pračky. Pomocí *Watch* máme možnost sledovat např. aktuální fázi či stav praní, teplotu, zvolený prací program apod.

[-] ◆ Casovac2	TON	local	
◆ IN	BOOL		TRUE
◆ PT	TIME		T#01s_500ms
◆ Q	BOOL		FALSE
◆ ET	TIME		T#800ms
◆ StartTime	TIME		T#50s_140ms
◆ M	BOOL		TRUE
◆ Restart	UDINT		0
◆ FazePrani	FazePrani_Type	global	HlavniPrani
◆ Stav	Stav_Type	global	Stav_RotaceBuben
◆ PraciProgram	PraciProgram_Type	local	Bavlna
◆ Otacky	Otacky_Type	local	OtackyDoprava
◆ KoleckoVolby	PraciProgram_Type	local	Bavlna
◆ AktualniTeplota	REAL	local	50.0
[-] ◆ VstupniHodnoty	VstupniHodnoty_Typ	local	
◆ PraciPrasekAktual	REAL		20.0
◆ Hmotnost	REAL		7.5
◆ OtackyNaStranu	INT		4
◆ OtackyDokonce	INT		23
◆ PraciPrasek	REAL		20.0
◆ PocetOtacek	INT		35
[-] ◆ Pracka	Pracka_Type	global	
◆ Napusteno	BOOL		TRUE
◆ Vypusteno	BOOL		FALSE
◆ Napousteni	BOOL		FALSE
◆ Vypousteni	BOOL		FALSE
◆ START	BOOL	global	TRUE

Obr. 1 Watch

5.2 Popis fází praní

Při spuštění pracího programu prochází prací cyklus těmito fázemi. Fáze Předpírka, Hlavní praní, Máchání a Odstřed'ování obsahují své stavy. Celý cyklus funguje jako stavový automat.

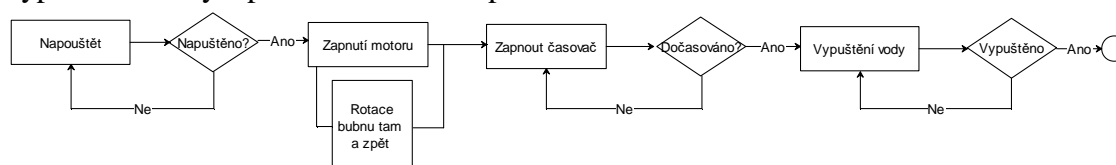


Nastavení - Před spuštěním pracího cyklu máme možnost zvolit si z několika pracích programů. K dispozici jsou programy Bavlna, Objemné, Vlna a Normální. Tyto jednotlivé prací programy mají přednastavené parametry pro další fáze praní, tzn., že při volbě některého z těchto programů budou některé fáze pracího programu trvat jinou dobu. Délku praní také ovlivňují vstupní parametry hmotnost prádla, možnost máchání navíc a také požadovaná teplota, která je v nastavení každého pracího programu, ale je možno ji změnit.

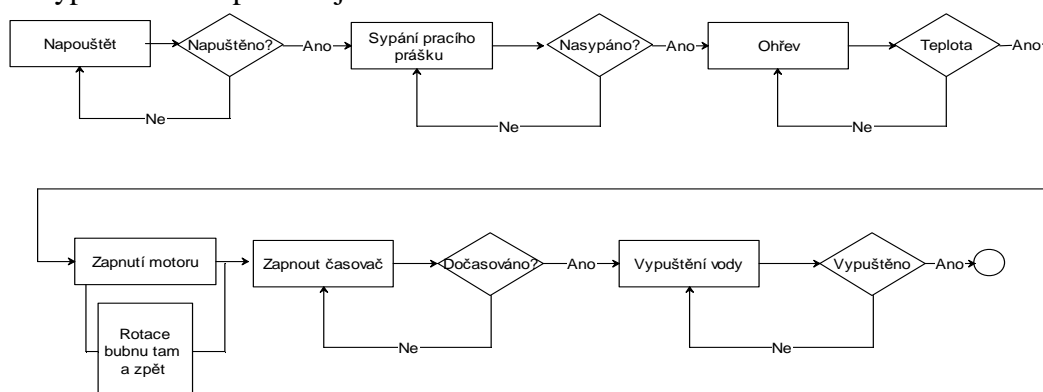
Start - Při stisku tlačítka START se začne vykonávat prací cyklus. Dokud je tlačítko START nezmáčknuté, tak zůstáváme v nastavení. Když zmáčkne START podruhé, tak se prací cyklus přeruší, při opětovném zmáčknutí pokračujeme v pracím cyklu tam, kde jsme skončili.

Zamčení dveří - Zamkne dveře.

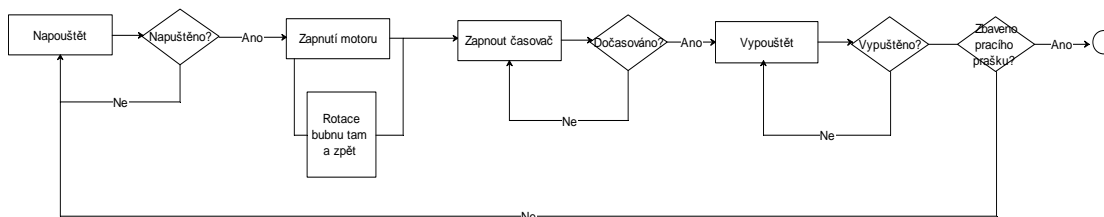
Předepírka - Slouží k odstranění velkých nečistot. Nejdříve se napustí voda a poté se zapne motor, který rotuje v určitých intervalech doprava a doleva. Délka rotace bubnu je závislá na pracím programu a hmotnosti prádla. Po dokončení rotace bubnu následuje vypouštění vody a přechod další fáze praní.



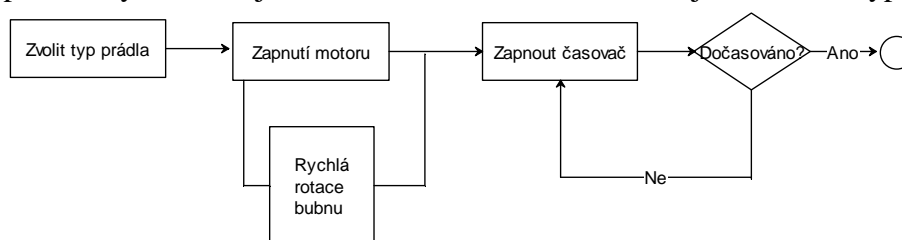
Hlavní praní - Vyčistí všechny zbývající nečistoty. Napustí se voda a poté se nasype prací prášek. Dále se ohřívá voda na požadovanou teplotu a po ohřátí začne rotovat buben doprava a doleva, tentokrát rotace ale trvá déle než ve fázi předepírka. Nakonec se vypustí voda a pokračuje se mácháním.



Máchání - Zbaví prádlo od pracího prášku. Napustí se voda, buben začne rotovat doprava a doleva a poté se voda vypustí. Pro jednoduchost je program udělán tak, že po každém máchání zůstane v prádle desetina pracího prášku a námi přijatelná hranice je jeden gram pracího prášku. Proto se máchání provádí pořád dokola, než je tato podmínka splněna. V případě, že jsme si v nastavení zvolili máchání navíc, tak po splnění podmínky se máchá prádlo ještě jednou.



Odstředování - Zhruba vysuší prádlo. Buben začne rotovat, tentokrát ale rotace probíhá rychle a na jednu stranu. Délka odstředování je závislá na typu prádla.



Uvolnění dveří - Uvolní dveře

Zvukový signál - Po dokončení pracího cyklu se na znamení konce praní ozve zvukový signál.

5.3 Popis klíčových částí implementace programu.

5.3.1 Rotace bubnu

Stav_RotaceBuben - tento stav obsahuj funkce `int predepirka()`, `int hlavni_prani()` `int machani()`. Vstupními parametry jsou `VstupniHodnoty.OtackyNaStranu` `VstupniHodnoty.PocetOtacek` a `VstupniHodnoty.Hmotnost`. Do proměnné `VstupniHodnoty.OtackyDokonce` bylo ve funkci `int nastaveni()` přiřazeno `VstupniHodnoty.PocetOtacek`. Na začátku tohoto stavu se rozběhne časovač, který časuje otáčky na určitou stranu, přičemž každá otáčka trvá v závislosti na `VstupniHodnoty.Hmotnost`. Jakmile časovač doběhne, tak se restartuje pro další průchod a buben se začne otáčet na druhou stranu a z proměnné `VstupniHodnoty.OtackyDokonce` se odečte `VstupniHodnoty.OtackyNaStranu`. Tento stav probíhá tak dlouho, dokud není hodnota `VstupniHodnoty.OtackyDokonce` nulová. V případě, že `VstupniHodnoty.OtackyNaStranu` nedělí `VstupniHodnoty.PocetOtacek`, tak v posledním průchodu bude zbývat méně otáček, než v `VstupniHodnoty.OtackyNaStranu`, a proto časovač musí časovat kratší dobu, což ošetřuje podmínka `if((VstupniHodnoty.OtackyNaStranu > VstupniHodnoty.OtackyDokonce) && (VstupniHodnoty.OtackyDokonce != 0))`, kde `PomocnaPromenna.pOtacky` je pomocná proměnná, která je po celou dobu nulová, jen v případě, že nastane uvedená podmínka, tak se do ní přiřadí hodnota. Jakmile je proměnná `VstupniHodnoty.OtackyDokonce` nulová, začne se provádět další stav.

```
case Stav_RotaceBuben:
{
    if((VstupniHodnoty.OtackyNaStranu > VstupniHodnoty.OtackyDokonce) && (VstupniHodnoty.OtackyDokonce != 0))
    {
        PomocnaPromenna.pOtacky = VstupniHodnoty.OtackyNaStranu - VstupniHodnoty.OtackyDokonce;
    }
    Casovac2.PT = (VstupniHodnoty.OtackyNaStranu - PomocnaPromenna.pOtacky) * VstupniHodnoty.Hmotnost * 50;
    Casovac2.IN = 1;
    if(Casovac2.Q)
    {
        Casovac2.IN = 0;

        switch(Otacky)
        {
            case OtackyDoprava:
                Otacky = OtackyDoleva;
                VstupniHodnoty.OtackyDokonce = VstupniHodnoty.OtackyDokonce - VstupniHodnoty.OtackyNaStranu;
                break;

            case OtackyDoleva:
                Otacky = OtackyDoprava;
                VstupniHodnoty.OtackyDokonce = VstupniHodnoty.OtackyDokonce - VstupniHodnoty.OtackyNaStranu;
                break;
        }
    }
    if(VstupniHodnoty.OtackyDokonce <= 0)
    {
        VstupniHodnoty.OtackyDokonce = VstupniHodnoty.PocetOtacek;
        PomocnaPromenna.pOtacky = 0;
        Pracka.Napusteno = 0;
        Pracka.Vypousteni = 1;
        Stav = Stav_Vypousteni;
    }
}
break;
```

Obr.2 Rotace Buben

5.3.2 Stav Napouštění

Stav_Napousteni - tento stav je taktéž obsažen ve funkcích `int predepirka()` `int hlavni_prani()` `int machani()`. `Pracka.Napousteni` a `Pracka.Napusteno` jsou globální proměnné, díky nimž komunikují programy *Pracka* a *Simulátor*. Jakmile zmáčkne START, tak se v programu *Pracka* ve funkci `int predepirka()` začne provádět napouštění pračky. Do proměnné *Stav* bylo přiřazeno v Init části *Stav_Napousteni*, tudíž se do proměnné `Pracka.Napousteni` přiřadí jednička a v programu *Simulátor* se začne provádět funkce `void napousteni()`. Až je buben naplněný vodou, tzn. časovač dočasoval, tak se do proměnné `Pracka.Napusteno` přiřadí 1, a v programu *Pracka* se začne provádět další Stav. Na konci funkcí `int predepirka()` `int hlavni_prani()` `int machani()` se do proměnné *Stav* přiřadí *Stav_Napousteni* a do proměnné `Pracka.Napousteni` jednička, aby se na začátku každé nadcházející funkce mohl provádět tento stav.

```
void _CYCLIC ProgramCyclic(void)
{
    if(Pracka.Napousteni)
    {
        napousteni();
    }
}

void napousteni()
{
    Casovac1.IN = 1;
    Casovac1.PT = 5000;

    if(Casovac1.Q)
    {
        Pracka.Napousteni = 0;
        Pracka.Napusteno = 1;
        Casovac1.IN = 0;
    }
}

int predepirka()
{
    switch(Stav)
    {
        case Stav_Napousteni:
            Pracka.Napousteni = 1;
            if(Pracka.Napusteno)
            {
                Pracka.Napousteni = 0;
                Stav = Stav_RotaceBuben;
            }
            break;
    }
}
```

The diagram illustrates the state transition logic between the *Program Simulator* and *Program Pracka*. In the *Program Simulator*, the `napousteni()` function sets `Casovac1.IN = 1` and `Casovac1.PT = 5000`. When the timer reaches zero (`Casovac1.Q` is true), it sets `Pracka.Napousteni = 0` and `Pracka.Napusteno = 1`. In the *Program Pracka*, the `predepirka()` function uses a switch statement on `Stav`. When `Stav` is `Stav_Napousteni`, it sets `Pracka.Napousteni = 1`. If `Pracka.Napusteno` is also 1, it resets `Pracka.Napousteni = 0` and changes `Stav` to `Stav_RotaceBuben`. Arrows indicate that `Pracka.Napousteni` is set to 1 in *Pracka* and then used in the *Simulator* to call `napousteni()`. Similarly, `Pracka.Napusteno` is set to 1 in the *Simulator* and used in *Pracka* to transition to the next state.

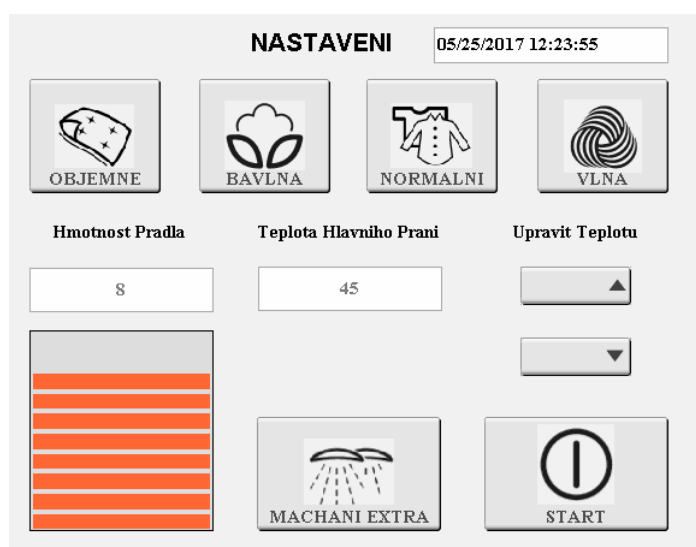
Obr.3 Program Simulator

Obr.4 Program Pracka

5.4 Vizualizace projektu

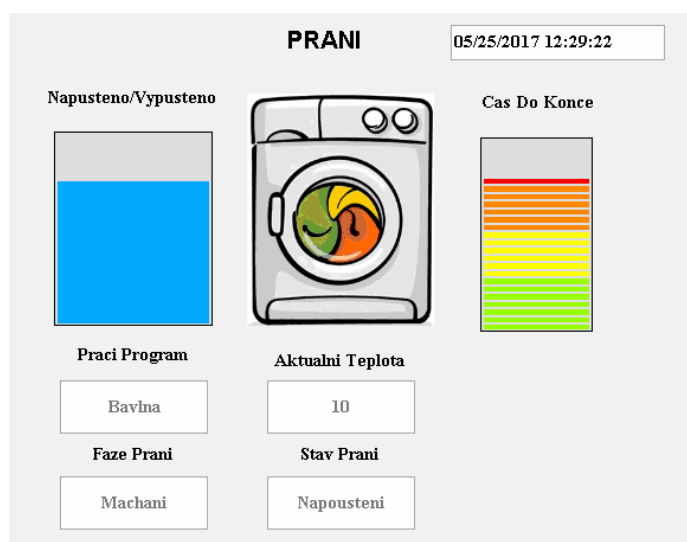
K usnadnění ovládání byla vytvořena jednoduchá vizualizace projektu. Vizualizace je spojená s proměnnými v programu, a proto je možné nastavovat nebo sledovat vybrané hodnoty.

Nastavení Pračky - Na úvodní straně máme možnost zvolit prací program, hmotnost prádla, změnit teplotu, anebo můžeme přidat máchání navíc. Tlačítkem START spustíme praní a dostaneme se na další stranu.



Obr.5 Nastavení

Praní - Druhá strana vizualizace slouží ke sledování fází praní a stavů, teploty, zvoleného pracího programu a času do konce praní.



Obr.6 Praní

6 ZÁVĚR

Úvodní dvě kapitoly byly věnovány představení, firmy B&R Automation a jejich technologií, neboť dané produkty jsou využívány v laboratorních cvičeních předmětu „Vyšší formy řízení“.

Další dvě kapitoly se zabývaly představením vývojového prostředí Automation Studio od firmy B&R Automation. V první části byly popsány hlavní vývojové nástroje a postup pro vytvoření projektu. V druhé části byly představeny programovací jazyky dostupné v Automation Studio, přičemž byly demonstrovány na jednoduché úloze *Osvětlení poschodí*.

Poslední a hlavní část bakalářské práce se zaměřila na realizaci programu do (modelu) automatické pračky. V této kapitole byla popsána analýza procesu praní, následně popis fází pracího cyklu, implementace klíčových částí programu a vizualizace daného projektu.

V příloze této bakalářské práce je dodán kompletní vývojový diagram a zdrojový kód pro daný projekt. K úloze *Osvětlení poschodí* jsou dodány zdrojové kódy ve všech programovacích jazycích. Nakonec byla vytvořena zadání úloh do cvičení, tato zadání jsou taktéž součástí přílohy.

7 SEZNAM POUŽITÝCH ZDROJŮ

- [1] B&R AUTOMATION [online]. 2012 [cit. 2017-05-24]. Dostupné z: <https://www.br-automation.com/cs/perfection-in-automation/>
- [2] B&R Eggelsberg. In: *STEFANZAUNER FOTOGRAFENMEISTER* [online]. 2011 [cit. 2017-05-25]. Dostupné z: <https://stefanzauner.wordpress.com/2012/03/10/modern-architecture-in-eggelsberg/br-eggelsberg-15/>
- [3] KAREL, Bílek. Automa.cz. *Automa* [online]. 2001, 1(3), 6 [cit. 2017-05-25]. Dostupné z: http://automa.cz/cz/casopis-clanky/ethernet-powerlink-komunikace-v-realnem-case-2001_03_33500_2769/
- [4] ZEŽULKA, František a Ondřej HYNČICA. *Průmyslový Ethernet VIII: Ethernet Powerlink, Profinet* [online]. Brno, 2008 [cit. 2017-05-25]. Dostupné z: http://www.uamt.feec.vutbr.cz/~zezulka/download/KPPA/A05_08s62_VIII.pdf
- [5] DUCHOSLAV, Petr. Desatero dobrých důvodů, proč investovat do robotů. *Factory Automation* [online]. 2014, , 4 [cit. 2017-05-25]. Dostupné z: <https://factoryautomation.cz/desatero-dobrych-duvodu-proc-investovat-do-robotu/>
- [6] B&R Industrial Automation. *B&R Industrial Automation - YouTube* [online]. 2016 [cit. 2017-05-25]. Dostupné z: www.youtube.com/user/BerneckerRainer/videos
- [7] ALEŠ, Vítek. Technologie reACTION od B&R posouvá balicí stroje na zcela novou úroveň. *T + T Technika a Trh* [online]. 2014, , 1 [cit. 2017-05-25]. Dostupné z: <https://www.technikaaatrh.cz/elektrotechnika/technologie-reaction-od-bar-posouva-balici-stroje-na-zcela-novou-uroven>
- [8] SLOVÁČEK, Antonín. *VIZUALIZACE ŘÍDICÍHO SYSTÉMU PRO INTELIGENTNÍ DŮM A JEHO OKOLÍ* [online]. Brno, 2010 [cit. 2017-05-25]. Dostupné z: https://www.vutbr.cz/www_base/zav_prace_soubor_verejne.php?file_id=30010
- [9] ČÁPKA, David. Úvod do programovacího jazyka C. *ITnetwork.cz* [online]. 4 [cit. 2017-05-25]. Dostupné z: <https://www.itnetwork.cz/cplusplus/cecko/zaklady/tutorial-uvod-do-programovaciho-jazyka-c>
- [10] VOJÁČEK, Antonín. Programovací režimy pro PLC dle IEC 61131-3 (CoDeSys). *Automatizace.hw.cz* [online]. 2011 [cit. 2017-05-25]. Dostupné z: <http://automatizace.hw.cz/programovaci-rezimy-pro-plc-dle-iec-61131-codesys>

- [11] Beckhoff New Automation Technology. *Beckhoff Information System* [online]. 2009 [cit. 2017-05-25]. Dostupné z: https://infosys.beckhoff.com/english.php?content=../content/1033/tcplccontrol/html/TcPlcCtrl_Languages%20IL.htm&id=
- [12] KOCHANÍČEK, Ludvík. COPTEL - Elektrotechnika. *COPTEL - Elektrotechnika* [online]. 2010 [cit. 2017-05-25]. Dostupné z: <https://coptkm.cz/portal/?doc=3905>

8 SEZNAM PŘÍLOH

CD obsahující: Vzorovou úlohu osvětlení poschodí v osmi programovacích jazycích
Projekt Automatická pračka

Vývojový diagram k projektu Automatická pračka
Úlohy do cvičení „Vyšší formy řízení“